ARI Research Note 87-72

DTIC FILE COPY

AD-A191 179

A COMPUTATIONAL MODEL OF MOTOR BEHAVIOR

Wayne Iba and Pat Langley
University of California at Irvine

at

Contracting Officer's Representative
Judith Orasanu

BASIC RESEARCH LABORATORY
Michael Kaplan, Director

U. S. Army
Research Institute for the Behavioral and Social Sciences

December 1987

Approved for public release; distribution unlimited.

U. S. ARMY RESEARCH INSTITUTE

FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

WM. DARRYL HENDERSON
COL, IN
Commanding

Research accomplished under contract
for the Department of the Army

University of California at Irvine

Technical review by

Dan Ragland

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARI Research Note 87-72	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A Computational Model of Motor Behavior		5. TYPE OF REPORT & PERIOD COVERED Interim Report March 86 - March 87
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Wayne Iba and Pat Langley		8. CONTRACT OR GRANT NUMBER(s) MDA903-85-C-0324
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Information and Computer Science University of California at Irvine Irvine, CA 92717		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q161102B74F
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Research Institute for the Behavioral and Social Sciences, 5001 Eisenhower Avenue, Alexandria, VA 22333-5600		12. REPORT DATE December 1987
		13. NUMBER OF PAGES 11
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* --		15. SECURITY CLASS. *(of this report)* Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE n/a

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

--

18. SUPPLEMENTARY NOTES

Judith Orasanu, contracting officer's representative

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Computational Models,
Artificial Intelligence
Cognitive Science
Motor Schema

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Generating even simple motor behavior using artificial manipulators has proven to be a very difficult task. A computational model of motor behavior is presented that assumes three inputs: a limb to carry out motor commands, a viewer-centered schema describing the desired behavior, and a sensory-motor interface allowing two-way communication between the agent and the environment. A motor schema is defined as a memory structure containing a few positions from the trace of a movement. The model then produces intervening points between those
(OVER)

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

ARI Research Note 87-72

20. Abstract (continued)

in the schema. A viewer-centered schema is transformed into its dual representation, facilitating execution of the desired movement on the limb. These two forms of a schema represent the same points in three-dimensional space, but behave in ways which have important differences when they are acted upon by the model. Our model accounts for a number of phenomena from the literature, including the speed/accuracy tradeoff, and the closed and open loop distinction. The model suggests directions for further experimentation.

A Computational Model of Motor Behavior

Wayne Iba
Pat Langley

Irvine Computational Intelligence Project
Department of Information & Computer Science
University of California, Irvine, CA 92717 USA

Abstract

Generating even simple motor behavior using artificial manipulators has proven to be a very difficult task. We present a computational model of motor behavior that assumes three inputs: a *limb* for carrying out motor commands, a *viewer-centered schema* describing the desired behavior, and a *sensory-motor interface* allowing two-way communication between the agent and the environment. We define a motor schema as a memory structure containing a few positions from the trace of a movement; the model produces intervening points between those in the schema. A viewer-centered schema is transformed into its dual representation, facilitating execution of the desired movement on the limb. These two forms of a schema represent the same points in three dimensional space, but behave in importantly different ways when acted upon by the model. Our model accounts for a number of phenomena from the literature including the speed accuracy trade-off and the closed loop and open loop distinction. The model suggests directions for further experimentation.

1. Introduction

All humans possess the ability to carry out skilled movements, but cognitive science has devoted insufficient attention to this important process. In this paper, we present a computational theory of motor behavior that is consistent with psychological studies of human motor skills. We view the model as filling an important gap between research on high-level cognitive processes (such as planning) and low-level neural mechanisms that control physical actions.

Psychological studies of motor behavior and skilled performance have been plentiful, but there has been little work addressing the representations and processes that underly such behavior. We believe that a computational theory of motor skills will lead to new insights about human motor behavior and suggest specific predictions that could be tested by further experiments. Such a theory should also help fill the gap between results in robotics and neuro-biological work on motor control.

In the following section, we briefly review several experimental results on human motor behavior, focusing on the psychological phenomena that our theory will address. After this we describe the theory, starting with its representational assumptions and then considering its basic mechanisms. Finally, we present some experimental results with the simulation of a two-jointed arm, observing the system's behavior as we change various parameters.

2. Human Motor Behavior

Space allows only a brief sketch of the nature of muscle control, but we direct the reader to Kelso [1982] for a more comprehensive treatment. In this section, we can only consider

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)*		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)*
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT *(of this Report)*		
17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*		
20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)* the basic structure of the human motor system, along with some of the major phenomena addressed by our theory.

Low-level motor control is handled by a servomechanism involving pairs of muscles that work in opposition. Maintaining the necessary forces of contraction in the respective muscles is performed by a local configuration of neurons and a muscle spindle operating in a feedback cycle. Increasing the load on a muscle will result in a greater contraction in the muscle fiber in response to the feedback from the muscle spindle. The important principle is that organisms are able to 'set' their limbs in a desired position relatively independent of loads and lack of visual or proprioceptive feedback. With this in mind, we now present the main phenomena which our model is able to account for.

One of the most robust findings in the motor literature is the tradeoff between speed and accuracy [Fitts et al.,1964; Langolf et al.,1976]. Humans can perform skills very accurately when carried out slowly, but their performance deteriorates as they 'run' the motion more quickly. This tradeoff gradually disappears as the skill is learned more completely with practice. Another finding is that the minimum time required to incorporate sensory feedback and initiate movements in response is approximately 200 msec. [Keele et al.,1968;Henry et al.,1961]. In the theory we propose, this limit is intimately related to the speed/accuracy tradeoff.

The literature on motor behavior makes a distinction between *closed loop* and *open loop* behavior [Stelmach,1982; Schmidt,1982a], but our framework creates a blend of the two. Closed loop behavior refers to motions that can be consciously adjusted as a result of the 'closed loop' of feedback. Open loop behavior describes movements that are performed without conscious control and without utilizing higher level feedback. Instead of using one model exclusively, we suggest there is a continuum from closed to open loop, where a point between the two is determined by the the amount of attentional resources that are required.

Empirical studies have also found a significant amount of skill transfer between different limbs. For instance, Hollerbach [1979] reports that handwriting generated by the nondominant hand is recognizably similar to that from the dominant hand. Such transfer occurs in other types of motor tasks as well, suggesting certain invariants in our representations of motor behaviors. We return to these phenomena, and our theory's explanation for them, after we have described our theory itself.

3. A Model of Motor Performance

Before describing our theory of motor behavior in detail, it is useful to clearly specify the inputs and outputs of the model. The first input is the structure of the limb involved in the skill; this may vary, but always consists of a set of connected joints, such as a shoulder, an elbow, and a wrist. Another input is the *motor schema* to be executed; this represents the intended movement. The final input is a *sensory-motor interface*; this provides the system with feedback about the results of its actions in the environment. The output of the model is motor behavior, described in terms of the locations and velocities of jointed limbs at successive points in time. Now let us turn to the representation of motor schemas in the theory.

3.1 Representing Motor Schemas and Motor Programs

Schmidt [1982b] has used the term *motor schema* to refer to some stored description of a motor skill. We will define this term more precisely as a sequence of points, $(X_1, X_2, \ldots, X_n)$, that describe the location of a connected set of joints (e.g., an arm) at successive time steps. Each point $X_i = (t, \{\langle J_k, \mathrm{P}, \mathrm{V}, m\rangle, \ldots\})$ contains a time value t along with a set of 4-tuples. The time indicates when (relative to the start of the schema) this particular set of 4-tuples should describe the condition of the joints. Each 4-tuple consists of (1) a joint label specifying a particular joint, (2) a desired position in three-space for the joint at time t, (3) a desired velocity vector describing the direction and speed for the joint upon reaching that position, and (4) a scalar multiple that is applied to the velocity as the joint leaves the position.

Within this framework, position and velocity vectors can be represented using two different coordinate systems. The *viewer-centered* representation simply uses Cartesian three-space with the origin at the base (the first joint) of the arm. This corresponds to the view an agent might receive while performing a skill.

An alternative scheme represents points in a *joint-centered* space, in terms of rotations about a local coordinate frame with the origin at each joint. That is, each joint has its own coordinate system in which location is represented using a distance ρ from the origin, an angle of rotation about the x-axis (θ_x), and an angle of rotation about the y axis (θ_y). The coordinate system for a particular joint J_i is defined in relation to the joint to which it is connected. Thus, the coordinates for an elbow would be described in the reference frame of its associated shoulder joint. This representation can be used to generate motor behavior; in addition, this is the form in which proprioceptive information is available during execution.

While these dual representations might seem unnecessary, they lend considerably to the model's explanatory power. We propose that a human would originally acquire a viewer-centered schema by observing another person performing a skill. Joint-centered schemas are created when an attempt to perform the observed task has been made. It is not immediately apparent that these two representations have quite different representational characteristics. While they are both able to specify any point in three-space, when acted upon by theory, they may produce quite different results. This results from the sparse structure of the motor schema. A viewer-centered schema can easily represent a straight line with two points, while the joint-centered schema would require an infinite number of points. This differential power of the two representations makes certain predictions about the types of tasks that would be easy or difficult to learn.

3.2 The Performance Component

Given a viewer-centered schema that describes some desired behavior, the performance system attempts to carry out this behavior using a specified limb. This involves a number of processes. First, the viewer-centered schema must be translated to a joint-centered representation. The resulting schema must then be 'run' by generating an executable motor program and carrying out the specified actions. Simultaneously, the agent must monitor the resulting states, comparing actual positions with the intended positions as given in the viewer-centered schema. Execution and monitoring proceed in parallel until an error is detected. At this point, the system initiates an error correction process to return the limb to the desired path. Now let us consider each of these processes in more detail.

From viewer-centered to joint-centered schemas. The first step in carrying out a motor skill involves translating the viewer-centered schema into a joint-centered representation that can be directly executed.[1] We will not consider the details of this transformation process, but we will assume that it is serial in nature and that it is potentially errorfull. Transformations must be done for each joint in a serial manner, starting with the base joint and considering each successive joint in turn. Errors may be introduced at each level, with small errors early in the process being compounded in more remote joints.

Executing the joint-centered schema. The joint-centered representation specifies only selected points involved in the skill. To actually generate behavior, one must have the desired locations and velocities for every joint at every point in time. We will use the term *motor program* to refer to such an interpolated schema. Motor programs are not stored in memory; they are generated in real time as the skill is executed. In our theory, the agent interpolates the points making up a motor program by generating a spline for each joint, connecting the sparser points in the joint-centered schema.[2] When the limb reaches the end of the first spline segment, the target point becomes the source and the next point in the schema becomes the target for the next spline. This method yields a smooth[3], continuous curve throughout the execution of the schema.

Monitoring. As we have seen, there is no guarantee that behavior generated by the joint-centered schema will correspond to that specified in the viewer-centered description. Thus, the agent must have some means of detecting divergences, and this is the role of the monitoring process. In order to make the necessary comparisons, the monitoring component uses the viewer-centered schema to generate a 'pseudo' motor program. This program cannot be executed by effectors, but it specifies the desired position at each time during execution. When the difference obtained from this comparison becomes noticeable (i.e., exceeds a threshold), the monitor interrupts execution and invokes the error correction process. The monitoring mechanism relies on two parameters: the frequency of monitoring and the error threshold.

Error Recovery. Once an error has been detected, the agent must recover from that error. When invoked by the monitoring process, the error recovery mechanism applies a 'burst of force' in a direction that will reduce the size of the error.[4]

---

[1] Such a schema may already exist in long-term memory, having resulted from earlier practice with the skill. In such cases, the skill can be executed without the transformation process, and thus occurs more quickly.

[2] We are making the assumption that low-level neural circuitry can take relatively sparse inputs from a schema and generate a motor program. Even if not completely accurate, this simplification can be corrected later, and in the meantime it lets us proceed with modeling the higher level aspects of motor skills.

[3] Movements which are not smooth, for example curves with a cusp, can also be generated with this method if so desired.

[4] This process models the type of conscious correction that results from error detection, and not an unconscious, servomechanism style of correction.

Error recovery involves generating a correction function that is added to points in the existing motor program. This correction function has an inverted U shape, starting with minor alterations, increasing to a peak, and then decreasing to zero after a time. The sum of the function's effects are equal to the size of the detected error. This means that if the error is constant, the path of the limb after error correction would return to the desired path after error correction has ended. The rate at which error correction occurs is controlled by a compensation parameter.

The use of an inverted U type correction function (sin, parabolic, or absolute value) causes a gradual change in the limb's actual movement over the lifetime of the correction process.[5] Depending on the circumstances, this method can produce undercorrections or overcorrections. The former occurs in cases where the uncorrected behavior was about to begin reconverging with the idealized path, but had barely exceeded the error detection threshold before this occurred. Since the original motor program would have returned to the desired path on its own, an overcorrection will result. In contrast, undercorrections will occur if the uncorrected behavior is still diverging from the desired path. Such cases will require multiple calls to the error recovery process.

4. Experimental Results

We have implemented our model of motor behavior as a running FranzLisp program. Although the theory is independent of the particular dimensions and rotational constraints, we have tested our system using a two-jointed arm with roughly human characteristics. Thus, the arm includes an upper arm and a forearm, the former rotating at a shoulder joint and the latter at an elbow joint. All of our tests have been run in two dimensions.

Initial tests have involved attempting to move the 'hand' through a straight line. Except for trivial cases, such motions are are extremely difficult for a jointed arm to execute [Hardy,1984], though they are easy to describe in viewer-centered coordinates. In a joint-centered representation, every movement of the arm must trace the path of an arc, and this can only be stored using many points closely spaced in time. In this way, the arm can be made to approximate a straight line by stringing together a sequence of many small arcs.

Figure 1 presents three pairs of curves from three separate execution trials.[6] The first of each pair are overlapped in the circular arc in the upper left corner of the figure because the lengths of the links are fixed as is the position of the shoulder joint. One of the curves represents the desired motion as described in the viewer-centered schema. Another pair traces the motion of the hand and elbow when we ran the model with *no* monitoring and error correction. The final pair of curves presents the resulting motion when monitoring and error correction were operational. As expected, the latter case approximates the ideal curve more closely than the monitoring-free case. However, neither behavior is a very good approximation. For this, the joint-centered schema would require additional points.

---

[5] Note that this introduces another parameter – the actual correction function. Along with this, we also include a parameter that specifies the duration of the correction process.

[6] Successive points were generated at equal time intervals, so one can estimate the velocities involved by examining the distances between points.

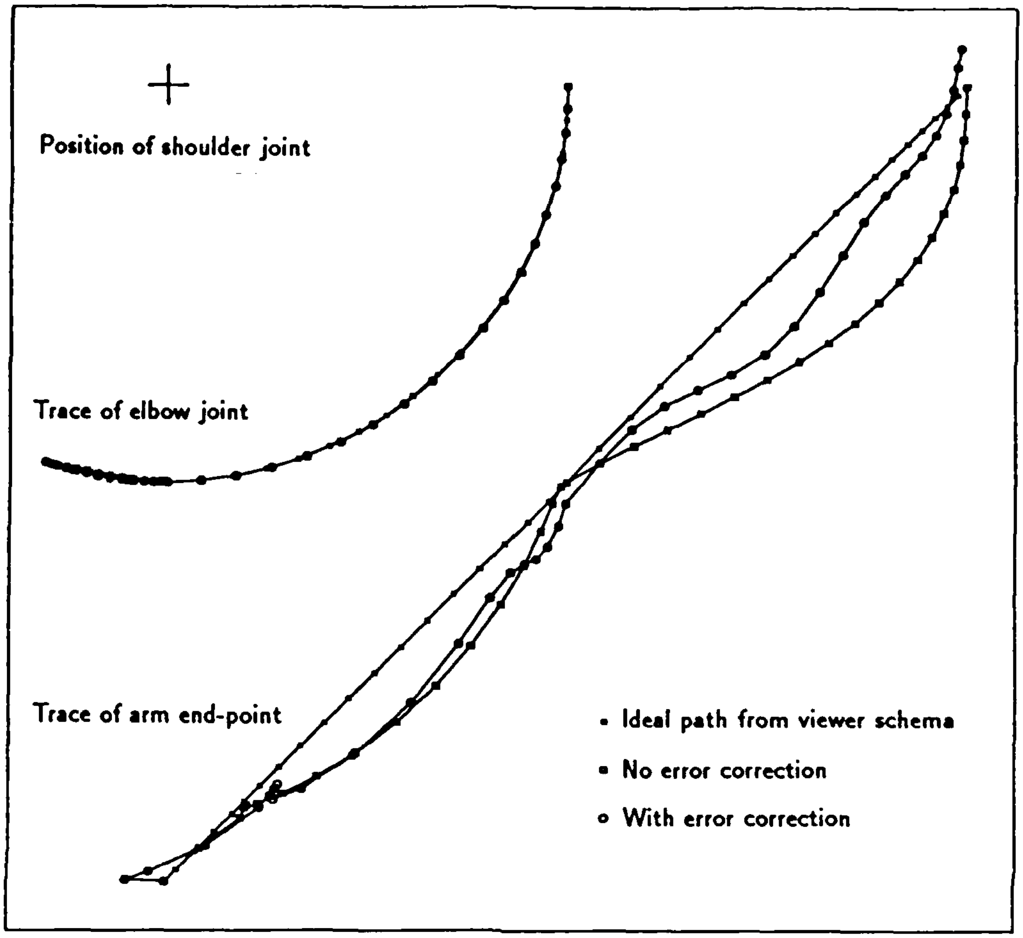

Figure 1
Two dimensional plots of three trials using a straight line schema.

Since we can run a schema at different speeds, we can test the model's ability to predict the tradeoff between speed and accuracy. Figure 2 shows that running the straight-line schema at higher speeds leads to greater deviations from the desired motion, i.e., to lower accuracy. We measure accuracy as the average deviation from the viewer-centered schema over the course of an execution. This score is computed by summing the angular deviation at each joint during monitoring and dividing by the total number of monitorings during that execution.

We have also noticed another intriguing regularity in the model's behavior. The implementation contains a parameter that scales the amount of compensation applied during error correction. Different settings of this parameter lead to different responses to error. Frequently the model detects an error as the deviation is becoming progressively greater, and radical corrective action is in order. However, such a remedy can also result in overcompensation, leading the model to 'overshoot' the desired position or trajectory.

Figure 3 presents the effects on the model's behavior as one alters the value of this parameter. When the schema is run slowly (making monitoring infrequent), increasing the

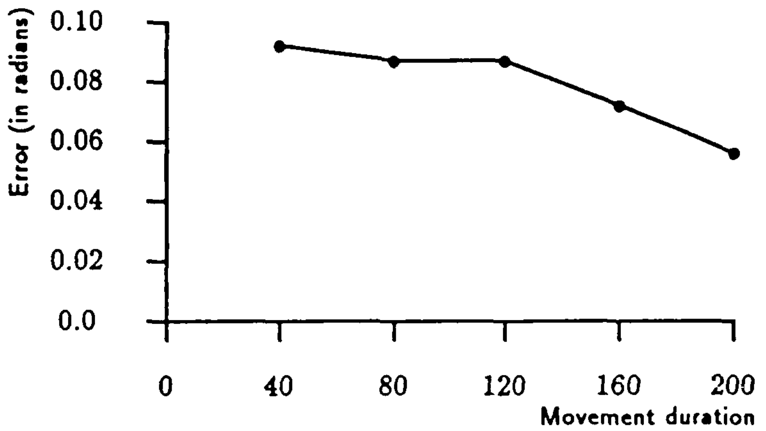

Figure 2
Decreasing error as a function of increasing movement time.

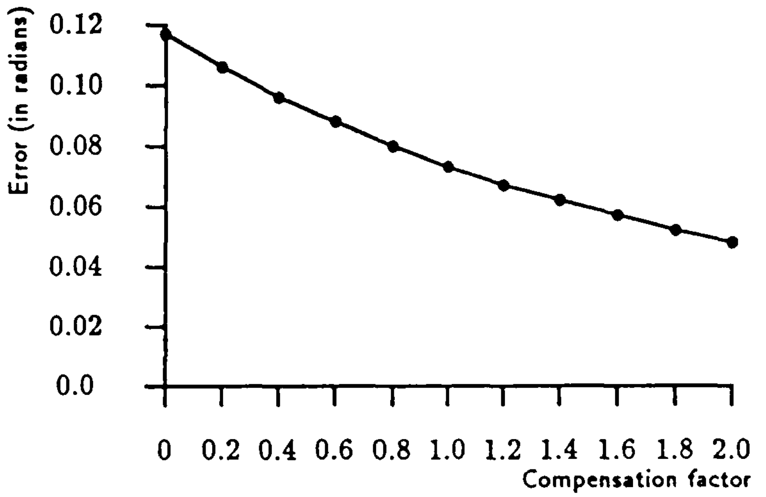

Figure 3
Decreasing error as a function of increasing compensation factor.

amount of correction leads to a decrease in the average deviation from the desired path. However, higher settings can actually produce worse performance when the schema is run slowly; for instance, when attempting to follow a straight line, the hand may instead follow a jagged line that cuts back and forth across the desired path. Although we certainly did not plan the model to behave in this fashion, we believe it makes sense. When monitoring occurs frequently, the system makes only minor errors and needs only minor corrective action. A high setting for the correction parameter will cause the system to overcompensate, and this can lead to wild oscillations.

In our performance model, we have seen only complete open or closed loop behavior. The trace of the execution without error correction in Figure 1 is an example of open loop behavior; the execution with error correction is an example of closed loop behavior. When a learning mechanism is included with the theory, improvements in the joint-centered schema will allow behavior to fall somewhere in between the two extremes, since less attention is required in order to maintain a given level of performance. We explain the transfer of motor skill between limbs by noting that our joint-centered schemas are stored independent of a particular limb. However, due to differences in the physical characteristics of the limbs, the resulting behavior would be deteriorated. Both of these phenomena are concerns for our continuing research.

5. Conclusion

In this paper we presented a computational theory of human motor behavior. The model assumes that two distinct representations underly motor skills, one based on viewer-centered coordinates and the other using joint-centered coordinates. Each consists of a sequence of 'points' that describe the locations and velocities of relevant joints at successive points in time. Motor behavior involves translating from the viewer-centered scheme to the joint-centered scheme, and then interpolating intermediate points to produce actual behavior.

We found that the two representations have different capabilities, each describing some motions better than the other. For this reason, the translation process is inherently imperfect and the agent must continually monitor his behavior for deviations from the desired path. When errors become noticeable, the agent invokes an error recovery process that attempts to put the bothersome joints back on track. We also argued that there was a lower limit to monitoring frequency, and that this limitation led naturally to the speed-accuracy tradeoff and the distinction between closed-loop and open-loop behavior.

Our initial tests of the model have been encouraging, but we need to consider its response given different motor schemas, with different numbers of points in each schema, and with different parameter settings. We also need to consider mechanisms for improving schemas as the result of experience. One obvious learning method would add a point to the joint-centered schema whenever its behavior required error recovery, but we plan to explore other techniques as well. Ultimately, we hope to develop a general and robust theory of human motor performance and learning.

Acknowledgements

This work was supported by Contract MDA 903-85-C-0324 from the Army Research Institute and by a gift from Hughes Aircraft Company. We would like to thank David Benjamin for typesetting the figures. The ideas in this paper have benefited from various discussions with the rest of the UCI World Modelers Group. Thanks also to Doug Fisher and Rogers Hall for comments on an earlier draft of this paper.

References

Fitts, P. M. & Peterson, J. R. (1964). Information capacity of discrete motor responses. *Journal of Experimental Psychlology,67*, 103-112.

Hardy, S. (1984). Robot control systems. In T. O'Shea & M. Eisenstadt (Eds.), Artificial intelligence: tools, techniques, and applications. New York, NY: Harper & Row.

Henry, F. M., & Harrison, J. S. (1961). Refractoriness of a fast movement. *Perceptual and Motor Skills,13*, 351-354.

Hollerbach, J. (1979). Understanding manipulator control by synthesizing human handwriting. In P. H. Winston & R. H. Brown (Eds.), *Artificial intelligence: an MIT perspective.* Cambridge, MA: MIT Press.

Keele, S. W., & Posner, M. I. (1968). Processing of visual feedback in rapid movements. *Journal of Experimental Psychology,77,* 155-158.

Kelso, J. A. S. (1982). Concepts an issues in human motor behavior: coming to grips with the jargon. In J. A. S. Kelso(Ed.), *Human motor behavior: an introduction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Langolf, G. D., Chaffin, D. B., & Foulke, J. A. (1976). An investigation of Fitt's Law using a wide range of movement amplitudes. *Journal of Motor Behavior, 8,* 113-128.

Schmidt, R. A. (1982a). More on motor programs. In J. A. S. Kelso(Ed.), *Human motor behavior: an introduction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Schmidt, R. A. (1982b). The schema concept. In J. A. S. Kelso(Ed.), *Human motor behavior: an introduction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Stelmach, G. E. (1982). Motor control and motor learning: the closed-loop perspective. In J. A. S. Kelso(Ed.), *Human motor behavior: an introduction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.